

COMMENTS

The enclosed is responsive to the Examiner's Office Action mailed on January 17, 2008. At the time the Examiner mailed the Office Action claims 1-3, 5-8, 10-13, 15-18, 20-23, 25-28 and 30 were pending. By way of the present response the Applicant has: 1) amended independent claims 1, 11 and 21; 2) not canceled any claims; and, 3) not added any claims. As such, claims 1-3, 5-8, 10-13, 15-18, 20-23, 25-28 and 30 remain pending.

The Examiner had maintained a rejection of the Applicant's independent claims under 35 USC 103 using primarily the Naganathan reference (U.S. Pub. Pat. No. 2004/0139194) in combination with the Novaes and Oliver references (U.S. Pat. No. 6,735,200 and U.S. Pub. Pat. No. 2003/0225876).

With the respect to the Applicant's earlier response filed 11/30/07 the Applicant stated (emphasis original)

The Applicant notes that Nagathan is directed to a system for "monitoring and measuring network service availability." See, Nagathan, para. [0012]. As far as the Applicant can tell, Naganathan is strictly limited to monitoring a network - not business logic software. As such, Nagathan does not appear to disclose any subject matter pertaining to "software components of a business logic process". Therefore the Examiner's analysis . . . is inaccurate at least so far as it concludes that Nagathan discloses monitoring of business logic software.

The applicant apologizes to the Examiner. The background discussion of Nagathan appears to discuss the monitoring of software for large enterprises. Therefore - at least for the moment - in order to proceed with a productive examination of the instant application, the Applicant is willing to go forward on the assumption that Naganathan's reference to a "network service" includes software programs that are made available through a network.

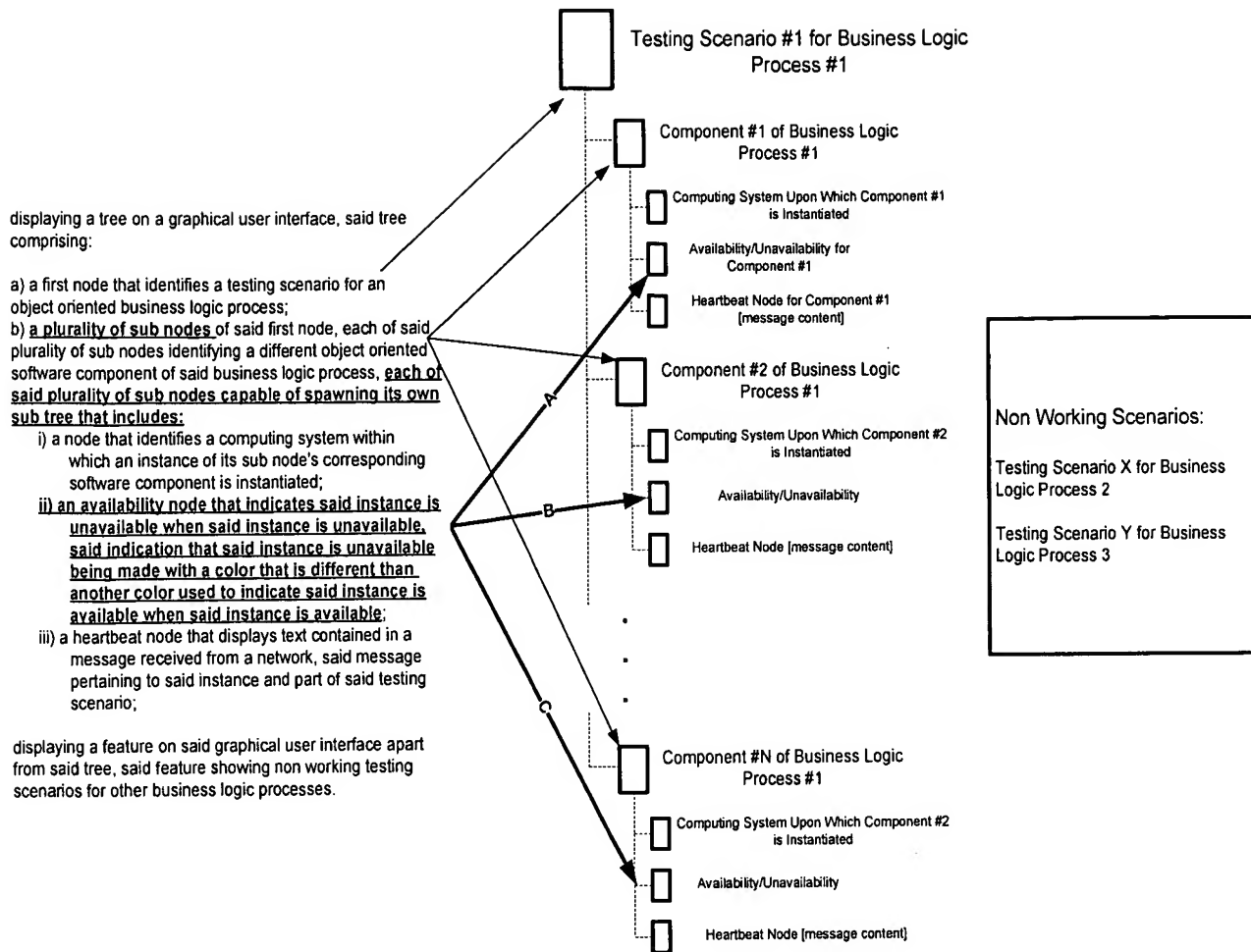
The Applicant also stated in the 11/30/07 Office Action response

[P]aragraph 0012 of Naganathan merely discloses a high level summary of the capabilities of the system disclosed by Naganathan. Stated simply, nothing in paragraph [0012] of Naganathan discloses the specifics of a GUI in sufficient detail to cover the claim element "... each of said one or more sub nodes capable of spawning its own sub node ...". The Applicant notes that this element is directed to the visual experience of a user of the GUI (i.e., the user can see a sub node spawned by another sub node). Paragraph [0012] of Naganathan is not directed to a GUI and cannot possibly disclose the specific GUI experience this claim element pertains to.

In response to the above argument, in the Office Action mailed on 1/17/08, the Examiner now points to paragraphs 0002, 0037 and the abstract of Naganathan as disclosing the claimed subject matter directed to the ability of a sub node rendered on a GUI to spawn its own sub node. Essentially the Examiner has taken the position that a statement made in Naganathan concerning a tree showing a hierarchical arrangement of components is sufficient disclosure. See, Examiner's Office Action, 1/17/08, pp. 6-7.

Rather than examine the Examiner's analysis statement-by-statement, the Applicant has instead opted to, immediately below, fully describe the teachings of the Naganathan reference and demonstrate how the Naganathan reference is insufficient to cover at least one pertinent feature of the Applicant's claimed subject matter. That is, rather than grinding out arguments with the Examiner that high level statements concerning "trees" and "hierarchies" are simply insufficient to cover the specific tree/hierarchy arrangement presently claimed, the Applicant instead demonstrates that the overall invention of Naganathan is not capable of tracking (and therefore not capable of rendering on a GUI) specific substantive information that is recited in the Applicant's claims.

Provided below for the Examiner's convenience is a diagram illustrating an embodiment of a GUI that would be covered by the Applicant's claims.



In particular, the Examiner's attention is drawn to element b) which has been amended in the present response to recite "a plurality of sub nodes". Arrows pointing out an example of a plurality of such sub nodes are indicated in the diagram above. Importantly, note that each of these sub nodes refer to a different component of a larger business logic process. That is, a first sub node is in reference to "Component #1 of Business Logic Process #1", a second sub node is in reference to "Component #2 of Business Logic Process #1" and an "Nth" sub node is in reference to "Component #N of Business Logic Process #1".

Thus, the Applicant's claim includes the limitation that smaller components of a larger business logic process are identified as separate nodes on the GUI presentation. (See Also, Fig. 4 of the Applicant's specification and corresponding discussion).

With element b) requiring separate nodes for smaller components of a larger business logic process, note that the structure and wording of the claim in conjunction with element ii) requires that the GUI show individual availabilities of smaller components of a larger business logic process. Corresponding arrows for this information are identified on the diagram above as arrows A, B and C. Thus, note that the exemplary GUI indicates the availability of "Component #1 of Business Logic Process #1", the availability of "Component #2 of Business Logic Process #1" and the availability of "Component #N of Business Logic Process #1". (See Also, Fig. 8 of the Applicant's specification and corresponding discussion).

The Applicant takes the view that the Applicant's specification is directed to a more global or expansive monitoring approach in which the respective availabilities of a business logic process's constituent components can be monitored and displayed on GUI as such. As the Applicant shows immediately below, the Naganathan does not provide for the kind of integration amongst software components or services that the Applicant's specification provides for. Therefore, the kind of information that is tracked and displayed according to the present invention is incapable of being tracked and displayed by Naganathan.

Simply stated, Naganathan can only indicate whether a service is available or unavailable. No insight into the effect that unavailability of a specific service might have on larger more complex business logic processes that depend on the service is provided.

Said another way, a user of Naganathan's invention is only informed that a service is unavailable, whereas, by contrast, a user of the Applicant's invention is informed that an entire business logic process that depends on a specific service is unavailable because the specific service is unavailable. This difference presents itself in the Applicant's claims, therefore, the Applicant's claims are patentable over Naganathan (the Applicant will also subsequently point out that Novaes and Oliver likewise fail to disclose this feature). Naganathan's failure to contemplate the availability tracking of a "service" as a larger business logic process's constituent component is detailed immediately below.

Naganathan indicates that the primary improvement of Naganathan over the prior art is the ability to determine service availability. Naganathan states (emphasis added):

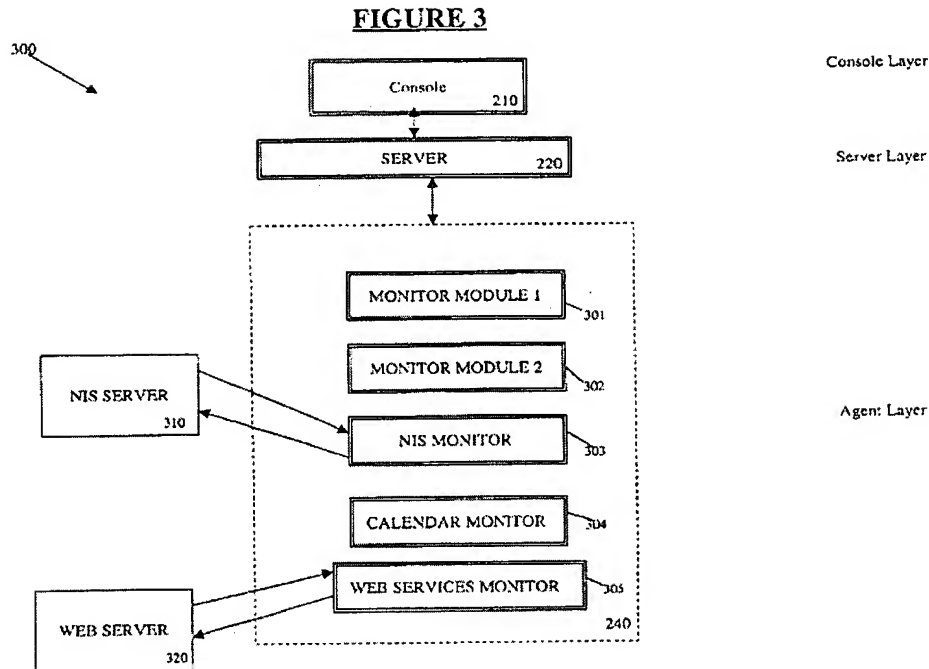
[0010] Although the prior art . . . provides the user with the convenience of accessing network resource services, the prior art system does not provide the user with the ability to monitor and measure network--service availability or unavailability.

According to Naganathan, the new-and-improved ability to monitor service availability stems from the introduction of system availability measurement and monitoring (SAMM).

According to Naganathan

[0038] The agent layer 230 further comprises a system availability measurement and monitoring (SAMM) system 240 of the present invention that allows users to specify network services they wish to measure and monitor. *The SAMM 240 can be dynamically loaded into the agent layer 230 by a user to perform the data gathering for the user to determine whether specified network services are available or unavailable.* The SAMM 240 provides service modules that are loadable by the agent 230 to monitor a particular network service.

The figure provided below depicts Figure 3 of the Naganathan reference.



In reference to the above diagram Naganathan states, in elaborating on the role of a SAMM (emphasis added):

[0044] FIG. 3 is a block diagram of one embodiment of the system availability measurement and monitoring system (SAMM) 240 of the present invention. As shown in FIG. 3, the SAMM 240 comprises a plurality of user loadable monitor modules 301-305, and service element module 310-320. In one embodiment of the present invention, the modules 301-305 are user loadable and they send periodic requests to the various services, which may be running locally or remotely, that a user wishes to monitor and measure in the network 200 and gather data. In the present invention, the modules 301-305 are referred to as synthetic modules. *Each of these synthetic transaction modules 301-305 has the ability to monitor and measure the availability of specific services from a user site. There is a separate module for each service that the user wishes to monitor. For example, the user may wish to monitor and measure the availability of a web service (e.g., module 305) or a calendar service (e.g., module 304).*

The Applicant respectfully submits that Naganathan only contemplates the availability testing of a service by itself - but does not contemplate the intertwining of such services

into a larger business logic process - or at least the ability to present on a GUI a cohesive image of such services as being smaller components of a larger business logic process along with their respective availabilities.

Here, the "services" being referred to by Naganathan are the types of components that an automated business logic process might depend upon (e.g., a web service, a calendar service, a DNS service (see, Naganathan para. [0041]), an NIS service (see, Naganathan para. [0047])). Such services therefore coincide with the Applicant's use of the term "components". For instance paragraph [0014] of the Applicant's specification recites:

[0014] Many business logic processes require a number of different software components to be invoked in a specific sequence. For example, an automated billing process might first run a database application to check the customer order records of the enterprise and then run an automated scripting application to create a custom tailored invoice for each order. Many business logic processes invoke a significant number of different software components over the course of their execution.

Thus, for instance, referring to the Applicant's example above of for an automated billing process, Naganathan appears to teach providing the availability status of the database application by itself and showing the availability status of the scripting application by itself but would not be able to present the availability status of a larger business logic process as a function of the availability of these lower level services.

As the Applicant sees it, Naganathan's primary failure is the inability to express any higher level relationship between the individual services whose availabilities are being tested for. If the Examiner chooses to identify Naganathan's service as a claimed "component", Naganathan fails to integrate and present these components as a larger

business logic process, contra-wise, if the Examiner chooses to identify Naganathan's service as a claimed "business logic process", Naganathan does not disclose the individual availability monitoring of the service's smaller constituent pieces. Moreover, again, Naganathan simply fails to discuss how availability information is presented on a GUI.

The Applicant also submits that the much disputed paragraph [0037] of Naganathan does not add anything more to what has already been discussed above concerning the Applicant's primary assertion that Naganathan does not disclose or render obvious the presentation on a GUI of the individual, respective availabilities of the components within a business logic process. Paragraph [0037] of Naganathan states:

[0037] In one embodiment of the present invention, the agents 230 use rule-based technology to determine that status of the managed objects. The agents 230 store data and status of managed objects in a management information base (MIB). The MIB acts a repository for managed objects. The managed objects together represent a model of the system and its components being managed. The managed objects are arranged in a tree, showing a hierarchical relationship of the components. Within the MIB, managed objects are logically grouped into management modules that collectively implement management functions.

More insight is provided into the meaning of paragraph [0037] by paragraph [0063] which states (emphasis added):

[0063] FIG. 7 is a computer implemented flow diagram illustration 700 of one embodiment of the network service availability monitoring and measurement system of the present invention. The network service availability monitoring and measurement of network services is initiated at step 705 upon initiation of the measurement system (SAMM 230) *initiating a management information base that stores the status of the managed objects on the network to which the SAMM 230 is connected.*

Taking the above in a light most favored to the Examiner's position, and without the Applicant admitting to as much, the most that the above paragraphs could be construed to mean is as follows: 1) a server's objects are organized into a MIB; 2) the organization and MIB is viewable on a GUI; and, 3) the SAMM connects into the MIB. Because the SAMM only inquires into the availability of an entire service - and not the affect that unavailability would have on an entire business logic process - the display on a GUI of a business logic process's individual components and their respective availabilities is simply not disclosed nor rendered obvious by Naganathan.

The Applicant respectfully submits that the Applicant's introduction of "object oriented" into the independent claims should be conclusive in this regard. If the Naganathan reference is to be deemed an object oriented software environment, the Applicant respectfully submits that a MIB provides object-level information. That is, one of ordinary skill would understand the MIB to present the organization of the individual objects within a server. The Applicant has submitted RFC 3444 in an enclosed IDS which supports this position. The information presented in the Applicant's GUI is understood to be at a much higher level. Simply put, each "component" in the Applicant's GUI represents perhaps large numbers of objects - and an entire business logic process is an even further collection of such components. Thus, the Applicant does not believe the MIB of Naganathan would track relationships between objects all the way up to the level of a business logic process nor would it be obvious to do so from the teachings of Naganathan.

Lastly, the Applicant respectfully submits that the Oliver and Novaes references are incapable of contributing to the Examiner's combination because these references are

directed to network monitoring and are therefore incapable of disclosing availability of software components within a larger business logic process presented as such on a GUI.

In the further interests of efficiency, the Applicant reserves the right under MPEP 2144.03.C to cause the Examiner to find in the prior art subject matter to which the Examiner has taken Official Notice at a later time in the prosecution of the present case when the subject matter of such prior art is actually at issue.

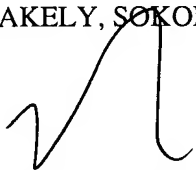
REMARKS

If there are any additional charges, please charge Deposit Account No. 02-2666. If a telephone interview would in any way expedite the prosecution of this application, the Examiner is invited to contact Robert B. O'Rourke at 408-720-8300.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: 4/17/08


Robert B. O'Rourke
Reg. No. 46,972

1279 Oakmead Parkway
Sunnyvale, CA 94085-4040
(408) 720-8300